Using the 21555 for System Slot Designs

Application Note

June 2001

Order Number: 278368-001

Contents

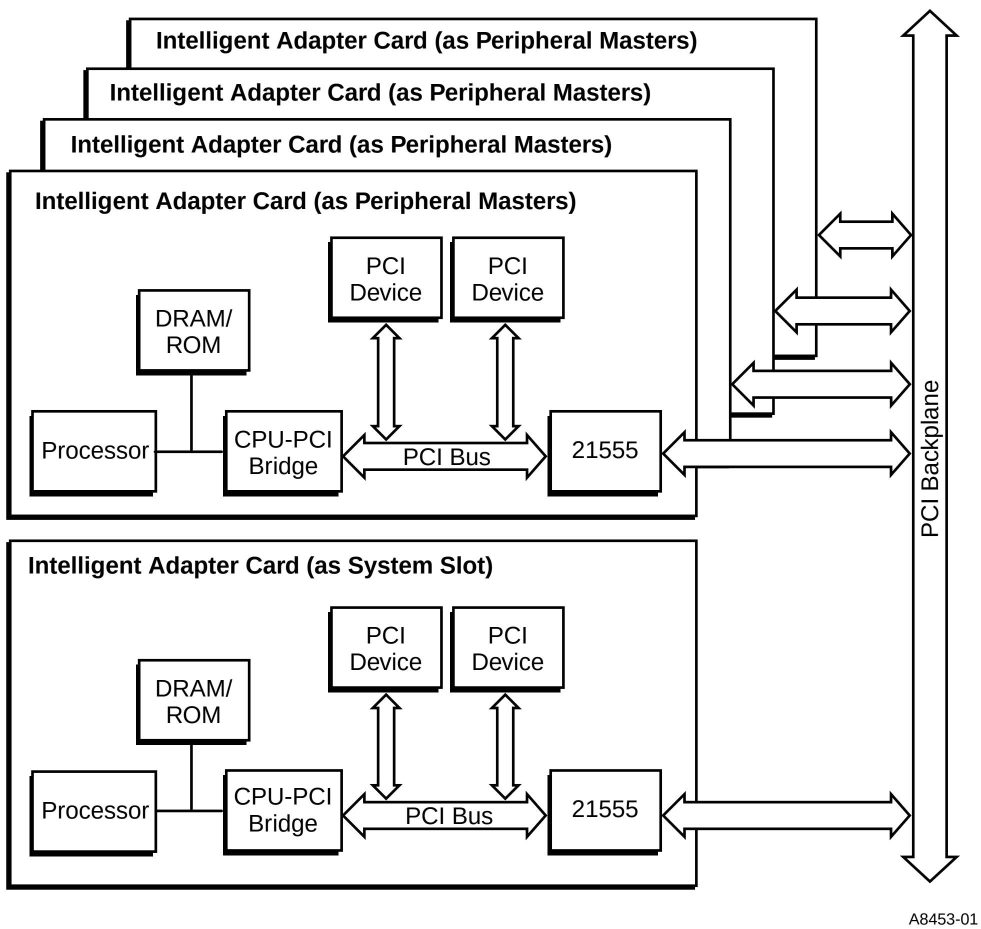
Figures

1.0 Introduction

This document describes how the 21555 resolves issues concerning the use of PCI add-in cards where the processor on the PCI add-in card is used as a system slot or host processor. It also describes 21555 issues and design considerations when the same add-in card is used as a peripheral master or as a local processor add-in card. This dual capability means that two separate add-in card design and development efforts are not needed to support system and peripheral slot add-in cards.

Figure 1 shows several intelligent PCI add-in cards plugged into the slots of a common PCI backplane. One or two slots in the PCI backplane are typically designated as system slots while the remaining slots are peripheral master slots. For example, CompactPCI* applications utilize this type of functionality.

Figure 1. PCI Backplane Application



Consider these issues if using the same add-in card as both a system slot and as peripheral master:

- Transparency versus isolation

  Transparency applies to any PCI add-in card that implements the 21555 when the processor is used as a system slot processor.
- Symmetry

  Symmetry is an issue when an add-in card performs as a system or as a peripheral master processor. The card slot symmetry depends on which slot the add-in card is plugged into.

1.1 Host System Slot Characteristics

These characteristic are needed for an intelligent PCI add-in card to be plugged into a system slot and to act as host processor:

- Configuration transparency to allow the host configuration of all slots. This allows transparent host access to the primary PCI interface of the peripheral master.
- Central function support for the backplane PCI bus.
  - — The assertion of the clock and reset generation, arbitration, and **REQ64#** during reset.
  - — The ability of the secondary interface to connect to the backplane to support central functions.

1.2 Local Peripheral Master Characteristics

These characteristic are needed for an intelligent PCI add-in card that is to be plugged into a peripheral master slot and to act as a local processor:

- Configuration and address map isolation of the subsystem to allow local processor control of local subsystem
- Asynchronous clocks
- The secondary interface connects to the local processor and the primary interface connects to the common backplane or toward the system slot.
- Hot Swap Support.

2.0 Transparent and Non Transparent Bridges

This section is an overview of transparent bridges and non transparent, or embedded bridges, and describes how they differ from each other. Refer to the 2*1555 Non-Transparent PCI-to-PCI Bridge User's Manual* for more detailed comparison of the two bridge types.

Transparent bridges are designed for expansion of slots and devices and to overcome electrical loading limits of the PCI bus. Transparent bridges assume a host processor is on the upstream or primary side of the bridge and that no intelligence is on the downstream or secondary side. These bridges provide hierarchical configuration support and a flat address map. Transparent bridges adhere to the *PCI-to-PCI Bridge Architecture Specification*.

The 21555 non transparent PCI-to-PCI bridge provides for isolation of the configuration and addressing functions. This isolation enables the local processor to control the subsystem.

The 21555 generates a secondary PCI bus that can support multiple PCI devices on PCI add-in cards and it provides PCI bus expansion. The 21555 assumes that there is a processor both upstream and downstream of the 21555. There may be transparent bridges between the processor and the 21555 on either side.

In the absence of a processor on one of the two interfaces, the 21555 secondary can be programmed by using a serial ROM on the secondary side or by using a primary side processor and the methods described in Section 3.0, “Transparency Versus Isolation” on page 7. A secondary side processor can also use the Section 3.0 methods to program the primary interface.

Note: The secondary PCI is not plug-and-play supported from the primary bus.

3.0 Transparency Versus Isolation

This section describes the conflicting isolation, and transparency requirements from a viewpoint of configuration and addressing.

3.1 Isolation of the Peripheral Master

The 21555 creates a configuration barrier by preventing the host system processor from enumerating buses or devices on its secondary side. This configuration barrier allows the local peripheral master processor to have complete control of the subsystem configuration without interference by the host system processor. To do this, the 21555 uses a Type 0 configuration register format.

When the standard BIOS or initialization code enumerates buses and devices, the initialization code is looking for a Type 1 configuration header which indicates the presence of PCI downstream buses and devices. When the initialization code encounters a Type 0 configuration header, it assumes that a PCI device was found and does not attempt to look beyond it.

A design cannot rely on plug-and-play initialization code to have a processor configure devices on the opposite side of the 21555, in either the upstream or the downstream direction. The ideal peripheral master configuration has the local processor connected to the secondary interface of the 21555 and the configuration registers are accessible to that local processor. Having the local processor perform both primary and secondary interface initialization requires device-specific code and thus, is not plug-and-play compatible. The 21555 does not respond to Type 1 configuration transactions.

All configuration registers can be accessed without the use of a local processor using host primary access by standard BIOS enumeration algorithms and serial ROM preload. The serial ROM preload can initialize all configuration registers that are not directly accessible from the host. This solution does not require device specific code and is useful for a peripheral master design that has no local processor. A device driver or custom BIOS can be used to configure the primary and secondary configuration registers from either interface using indirect configuration cycles. Self - addressed indirect configuration cycles are used to access the opposite side of the 21555 for configuration transactions. A local processor or Serial ROM is not required in this case but the host must use device-specific code to properly initialize the 21555.

3.2 An Alternative Peripheral Master Design

The single design for peripheral master and system slot add-in cards is an alternative for the peripheral master design. This design is physically configured with the processor chipset that is connected on the primary side of the 21555 and the secondary side that is connected to the backplane. For the peripheral master in this design, the 21555 central function, secondary arbitration, secondary clock outputs, and primary lockout features are all disabled using strapping. The Hot Swap signals and registers are synchronized to the primary clock but connected to the backplane (secondary) and treated as asynchronous.

Software configuration of the 21555's secondary configuration registers with no local processor on the secondary bus requires the use of indirect configuration cycles from the local processor or from the system slot processor with the use of the generic own bits for atomic access.

3.3 System Slot Design

The ideal system slot design has the bridge orientated the same as the alternative peripheral master design, with the secondary of the 21555 facing the backplane. In this orientation, a mechanism does exist in the 21555 for hierarchical configuration of downstream or upstream devices but device-specific initialization code is needed. The 21555 implements device specific registers that allow the 21555 to initiate configuration transactions on the opposite interface, upstream or downstream, using any address or data. Refer to the *21555 Non-Transparent PCI-to-PCI Bridge User's Manual* for details on this feature.

Note: A general bus scan using the indirect configuration mechanism is possible. The 21555 is programmed to respond to configuration transaction that it initiates on either the primary or the secondary PCI bus. Therefore, the indirect configuration mechanism can now be used to access a 21555 configuration register from the opposite interface. Be advised that worst case analysis is required to avoid problems on the local PCI bus with the 21555 claiming its own configuration cycles. See *PCI Local Bus Specification Revision 2.2,Item 9 of Section 3.10* entitled *Special Design Considerations*.

The system slot (host) processor has direct access to all 21555 configuration registers that are writable from the primary side. Some configuration registers are restricted from direct host primary configuration write accesses. By using indirect configuration cycles and by the 21555 claiming its own configuration cycles, it is possible to indirectly access the read only configuration registers from the primary side of the 21555.

The 21555 allows the system slot configuration mentioned in Section 1.0, "Introduction" on page 5 with advanced features. Configuration transparency is achieved through the use of indirect configuration cycles issued from the primary or secondary side. This allows host configuration of all slots from the primary or secondary sides. Using a single design for system slot or peripheral slot it is recommended that the processor be on the primary side of the 21555 and that the secondary side creates the PCI system slots. Discovery and Configuration must be done entirely with indirect configuration cycles. The secondary interface of the 21555 provides central function support for the backplane PCI bus, clock generation, arbitration, reset generation, and **REQ64#** assertion during reset.

3.4 Addressing

The 21555 creates independent address maps for both the primary and secondary bus domains. The 21555 uses base address registers (BARs) to designate both downstream and upstream forwarding windows. Other device-specific registers hold address translation information. Typically, the serial ROM or local processor sets up the size and type of these BARs. The local processor also sets up the address translation registers and the map the BARs on the secondary interface used for upstream forwarding. The host processor is typically expected to only map the BARs on the primary interface, using plug-and-play BIOS or initialization code after local processor setup. However, in the application described here where the secondary interface of the 21555 faces the backplane, device-specific code on the host processor is needed to initialize the setup registers, address translation registers, and BARs for the secondary interface. Essentially, the host processor on the system add-in card must also perform the local processor duties for the 21555 from the primary side.

System slot designs can also be implemented in a plug-and-play compatilble design using standard transparent PCI-to-PCI bridges. A standard transparent PCI-to-PCI bridge uses a flat address map, which is used for all PCI devices on either side of the bridge. Address windows are defined in the transparent PCI-to-PCI bridge for downstream transaction forwarding. Inverse decoding is used to forward upstream those transactions with addresses that fall outside of these windows. No address translation is used. Standard BIOS or initialization code automatically sets up these registers.

4.0 Symmetry Issues

The 21555 assumes that a host processor is present on the primary interface and that a local processor is present on the secondary interface. Implementing the 21555 as both a peripheral master and as system add-in card implies that:

- The 21555’s primary interface on the system add-in card connects to the host processor.
- The secondary interface connects to the backplane.

Therefore, none of the 21555 devices has a processor on the secondary side since all connect to the backplane, and the host processor of this configuration is attempting to control and configure the system from the primary side of the 21555.

This configuration creates issues around asymmetries that exist between the primary and secondary interfaces in the 21555. There are asymmetric features in the 21555 architecture. These features are application dependent.

4.1 Reset

The 21555 cannot assert the primary reset signal and does not sample the secondary reset signal.

The reset signal flows from the primary interface to the secondary interface. The primary reset, **p_rst_l**, is an input signal, the secondary reset, **s_rst_l**, is an output signal. The 21555 assumes a host-driven reset. When the host system is reset, the subsystem is reset. There are other reset mechanisms, including a secondary reset bit and a chip reset bit, that also assert the secondary reset signal. The 21555 also has a **s_rst_in_l** pin that is used to assert **s_rst_l** and has the same effect internally to the 21555 as asserting **p_rst_l.**

4.2 Configuration Lockout

The 21555 allows local processor configuration from the secondary interface before a host access is allowed by locking out the primary interface during this time. Primary bus transactions receive a target retry. This primary lockout is terminated by clearing the primary lockout bit through the serial ROM preload or by local processor access to a configuration register containing the primary lockout bit. It is disabled by a strapping option during the rising edge of **p_rst_l**.

4.3 I2O Controller

The 21555's I2O controller is oriented so that the IOP (I/O controller, or local processor) is on the secondary side. The host processor is on the primary side. The I2O controllers are only used when the add-in card is used as a peripheral master. The 21555 I2O controller on the system add-in card must be disabled. Refer to the *21555 Non-Transparent PCI-to-PCI Bridge User Manual* for more information.

4.4 Clocks

Clocks are only an issue if the secondary clock output is used; otherwise the clock output is disabled and the independent primary and secondary clocks are used. Signal **s_clk_o** is a buffered version of the primary clock **p_clk**. When used, **s_clk_o** is externally buffered and is used for all secondary bus devices, including the 21555's secondary interface.

4.5 Arbiter

The 21555 has an internal arbiter for the secondary bus. This arbiter can be disabled and an external arbiter can be used. An external arbiter is difficult to construct in PALs, and generally require additional loads on the **FRAME#**, **IRDY#**, **CLK**, and **RST#** signals. An external arbiter is needed for the primary interface.

4.6 Central Functions

During reset, the 21555 drives the secondary signals **AD[31:0], C/BE#[3:0]**, and **PAR** to zero (0), if the 21555 is configured as the central function. The 21555 asserts the secondary signal **REQ64#** during the secondary bus reset. These functions can be disabled and performed externally.

The 21555 does not perform these functions on the primary bus. An external agent must assert the **REQ64#** signal on the primary bus during primary bus reset to enable the 64-bit interface. During the assertion of the **s_rst_in_l** signal, a strapping option exists to force the primary bus into 64 bit mode.

4.7 Expansion ROM BAR

Parallel ROM access through the expansion ROM BAR register is supported only in the primary interface configuration register set. The only way that the parallel ROM is accessed by the secondary interface is through device specific CSRs. An expansion ROM is only used when the add-in card is a peripheral master and is not used when it is a system add-in card.

It is recommended that an expansion ROM be used only for peripheral master add-in cards and not be used for system add-in cards.

4.8 DAC Forwarding

The 21555 uses inverse decoding and does not perform address translation above the 4GB boundary. There is a 64-bit BAR for downstream forwarding, and the 21555 forwards all secondary bus DAC transactions outside of this range upstream. If 64-bit addressing is not used, this is not an issue. Otherwise, set up 64-bit addressing windows carefully.

4.9 CompactPCI* Hot Swap

The 21555 assumes that the primary interface connects to the add-in card edge. Upon insertion or removal, the 21555 design assumes that the local reset is ORed on the board with the primary bus reset and input to the **p_rst_l** (primary reset input) pin. The 21555, in turn, asserts **s_rst_l** to reset the subsystem. The enumeration notification interrupt signal, **p_enum_l**, is synchronized to the primary bus clock. Designs using the secondary side of the 21555 to create a system PCI bus in a Hot Swap design may treat the HSCSR (Hot Swap control and Status register) **l_stat**, and **p_enum_l** as asynchronous.

4.10 Power Management

21555 power management **PME#** support is designed with **s_pme_l** as an input signal on the secondary side and **p_pme_l** as an output signal on the primary side. Also, the 21555 interrupts a secondary bus interrupt (**s_inta_l**) to inform the subsystem that it has been moved from either the D1 or D2 power state to the D0 power state.

4.11 BAR variations

The 21555 primary and secondary base address register vary in the following ways:

- The first BAR at location 10h on the primary side is configured to forward downstream memory transactions in addition to mapping 21555 CSRs (primarily to support I2O functionality). The corresponding BAR on the secondary side maps the 21555 CSRs only.
- The primary side implements a BAR that is configured as a 64-bit BAR, the secondary does not (see Section 4.8).
- The secondary side implements a BAR that uses lookup table base address translation, the primary side does not.

4.12 Class Codes

The primary interface class code configuration register is preloaded to reflect a vendor specific value. This class code is used to give the subsystem its class identity (for example, storage, network, and so on). The secondary side class code reads as “bridge, other” and cannot be modified.

5.0 Conclusion

With device-specific code and perhaps some special hardware, it is possible to create a add-in card using the 21555 that is versatile enough to function as a system and a peripheral master. The standard PCI-to-PCI Bridge has the transparency that the system prefers, both in terms of a plug-and-play hierarchical configuration and a flat addressing model. By using a separate system add-in card, the primary interface is connected to the host processor and the secondary side to the add-in card edge. This avoids many of the symmetry issues and takes advantage of clocking, arbitration, and central functions that are provided on the secondary interface. A standard PCI-to-PCI bridge has more stringent requirements for having the host processor on the primary side of the bridge.

The 21555 is particularly suited for peripheral master add-in cards where subsystem configuration and addressing isolation is desired. By using two separate designs with opposite 21555 orientation or a transparent bridge on the system slot, design symmetry issues are avoided between a peripheral master add-in card and system slot add-in cards. A single design is possible but requires more complex device dependent BIOS and drivers.

The single design for both add-in cards with a processor chipset present to interface the processor to the PCI bus must have the central function, loading, arbitration, clocking, and **REQ64#** at reset to manage the devices on the primary side of the 21555.

There are trade-off decisions between utility, cost, redundancy and software development to be considered when finalizing a 21555 design to decide which is the best alternative in each case.